



El futuro digital  
es de todos

MinTIC

# Unidad 1

## 2 - Clases útiles en Java Git y Github





# Clase String

- Un String en Java representa una **cadena de caracteres no modificable**.
- Todos los literales de la forma "*cualquier texto*", es decir, literales entre comillas dobles, que aparecen en un programa java se implementan como objetos de la clase String.
- Se puede crear un String de varias formas, entre ellas:

- Utilizando una cadena de caracteres entre comillas:

```
String s1 = "abcdef";
```

- Utilizando operador de concatenación + con dos o más objetos String:

```
String s2 = s1 + "ghij";           //s2 contiene "abcdefghij"
```

```
String s3 = s1 + s2 + "klm";      //s3 contiene  
"abcdefabcdefghijklm"
```



# Clase String



El futuro digital  
es de todos

MinTIC

Método	Descripción
<code>length()</code>	Devuelve la longitud de la cadena
<code>indexOf('caracter')</code>	Devuelve la posición de la primera aparición de carácter dentro del String. Devuelve -1 si no lo encuentra.
<code>lastIndexOf('caracter')</code>	Devuelve la posición de la última aparición de carácter dentro del String. Devuelve -1 si no lo encuentra.
<code>charAt(n)</code>	Devuelve el carácter que está en la posición n
<code>substring(n1,n2)</code>	Devuelve la subcadena desde la posición n1 hasta n2 - 1
<code>toUpperCase()</code>	Devuelve la cadena convertida a mayúsculas
<code>toLowerCase()</code>	Devuelve la cadena convertida a minúsculas
<code>equals(otroString)</code>	Compara dos cadenas y devuelve true si son iguales
<code>equalsIgnoreCase(otroString)</code>	Igual que equals pero sin considerar mayúsculas y minúsculas
<code>valueOf(N)</code>	Convierte el valor N a String. N puede ser de cualquier tipo.
<code>trim()</code>	Devuelve la copia de la cadena, eliminando espacios en blanco en ambos extremos. No afecta los espacios en blanco en el medio.



# Clase Math

- La clase **Math** nos ofrece numerosos y valiosos métodos y constantes estáticos, que podemos utilizar tan sólo anteponiendo el nombre de la clase, (**Math.metodo()** ó **Math.CONSTANTE**).
- Hay muchas funciones disponibles. Se puede consultar la lista completa en la documentación oficial del [API](#) de Java

Constante	Descripción
E	Devuelve el valor más aproximado del número e
PI	Devuelve el valor más aproximado del número PI



# Clase Math



El futuro digital  
es de todos

MinTIC

Función	Significado	Ejemplo	Resultado
<code>abs()</code>	Valor absoluto	<code>int x = Math.abs(-2.3);</code>	2.3
<code>sin()</code>	Seno	<code>double x = Math.sin(0.5);</code>	0.4794255386042
<code>cos()</code>	Coseno	<code>double x = Math.cos(0.5);</code>	0.87758256189037
<code>tan()</code>	Tangente	<code>double x = Math.tan(0.5);</code>	0.54630248984379
<code>exp()</code>	Exponenciación neperiana	<code>double x = Math.exp(1);</code>	2.71828182845904
<code>log()</code>	Logaritmo neperiano	<code>double x = Math.log(2.7172);</code>	0.99960193833500
<code>pow()</code>	Potencia	<code>double x = Math.pow(2, 3);</code>	8.0
<code>round()</code>	Redondeo	<code>double x = Math.round(2.5);</code>	3
<code>floor()</code>	Aproximación entero mayor	<code>double x = Math.floor(2.5);</code>	2
<code>ceil()</code>	Aproximación entero menor	<code>double x = Math.ceil(2.5);</code>	3

# Clases Wrapper

- En ocasiones es muy conveniente poder tratar los datos primitivos (int, boolean, etc.) como objetos. Pero los datos primitivos no son objetos, con lo que quedan en principio excluidos de estas posibilidades.
- Para resolver esta situación el API de Java incorpora las clases envoltorio (wrapper class), que no son más que dotar a los datos primitivos con un envoltorio que permite tratarlos como objetos.

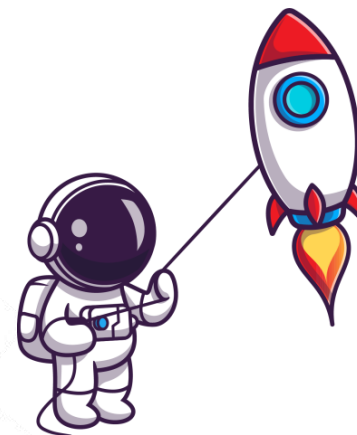
Clase	Tipo primitivo
Byte	byte
Short	short
Integer	int
Long	long
Float	float
Double	double
Boolean	boolean
Character	char





# Clase System

- El uso de la clase `System` es algo de utilizamos todos, normalmente invocando `System.out.println("hola")` o algo muy similar.
- Dispone de las variables estáticas `in`, `out` y `err` que hacen referencia a la entrada, salida y manejo de errores respectivamente.
- La clase `System` tiene métodos muy útiles ya que es la encargada de interactuar en el sistema.







El futuro digital  
es de todos

MinTIC

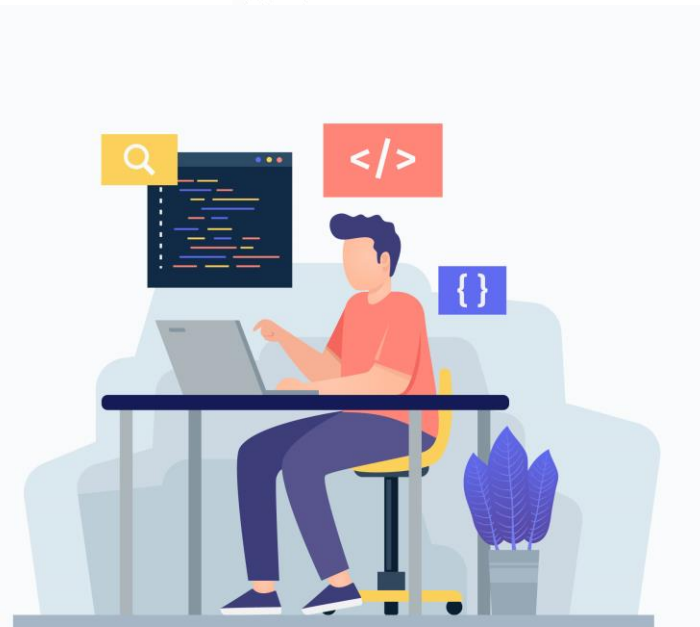
# Ejercicios





# Vamos al código

- Crear un proyecto Maven para los ejercicios de la clase 2
  - Ctrl + Shift + P
  - > Java: Create Java Project...
  - Maven
  - *maven-archetype-quickstart*, <versión más reciente>
  - **group Id:** co.edu.utp.misiontic2022.c2
  - **artefact Id:** clase02
- Por cada ejercicio propuesto, crear una función estática (método) que resuelva dicho ejercicio.
- Usar la función `main()` solo para hacer el llamado a las funciones que se quieren probar.



[Vector de Tecnología creado por freepik - www.freepik.es](https://www.freepik.es/fotos-vectores-gratis/tecnologia)



# Ejercicios

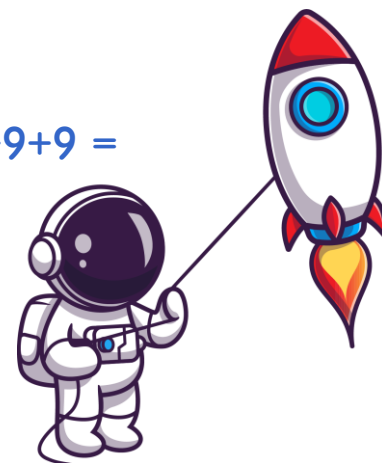
1. Programa que pida por teclado la fecha de nacimiento de una persona (día, mes, año) y calcule su número de la suerte.  
El número de la suerte se calcula sumando el día, mes y año de la fecha de nacimiento y a continuación sumando las cifras obtenidas en la suma.

**Por ejemplo:**

Si la fecha de nacimiento es 12/07/1980

Calculamos el número de la suerte así:  $12+7+1980 = 1999$   $1+9+9+9 = 28$

Número de la suerte: 28





# Ejercicios

2. Programa que calcule el precio de venta de un producto conociendo el precio por unidad (sin IVA) del producto, el número de productos vendidos y el porcentaje de IVA aplicado. Los datos anteriores se leerán por teclado.
3. Programa que lea dos variables enteras  $N$  y  $m$  y le quite a  $N$  sus  $m$  últimas cifras.

Por ejemplo, si  $N = 123456$  y  $m = 2$  el nuevo valor de  $N$  será 1234.

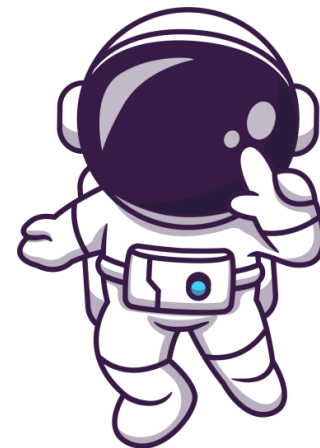
4. Programa que pase una velocidad en Km/h a m/s. La velocidad se lee por teclado.
5. Programa que lea la longitud de los catetos de un triángulo rectángulo y calcule la longitud de la hipotenusa según el teorema de Pitágoras.





# Ejercicios

6. Programa que lea un número entero y muestre si el número es múltiplo de 10.
7. Programa que lea un carácter por teclado y compruebe si es una letra mayúscula.
8. Programa que lea dos números por teclado y muestre el resultado de la división del primer número por el segundo. Se debe comprobar que el divisor no puede ser cero.
9. El programa lee por teclado tres números enteros y calcula y muestra el mayor de los tres.





# Ejercicios



10. Programa que lea por teclado tres números enteros H, M, S correspondientes a hora, minutos y segundos respectivamente, y compruebe si la hora que indican es una hora válida.
11. Programa que lea una variable entera mes y compruebe si el valor corresponde a un mes de 30 días, de 31 o de 28. Supondremos que febrero tiene 28 días. Se mostrará además el nombre del mes. Se debe comprobar que el valor introducido esté comprendido entre 1 y 12.
12. Realizar programa que muestre los números del 1 al 100 que no sean múltiplos de 3, utilizando cada una las instrucciones repetitivas (while, do while, for)



# Ejercicios

13. Programa que pase de pesos a dólares y viceversa.
14. Mostrar los N primeros términos de la serie de Fibonacci.
15. Leer un número N y calcular la suma de los factoriales de los números desde 0 hasta N.
16. Programa que muestre en líneas separadas lo siguiente:

ZYWXVUTSRQPONMLKJIHGFEDCBA

YWXVUTSRQPONMLKJIHGFEDCBA

WXVUTSRQPONMLKJIHGFEDCBA

....

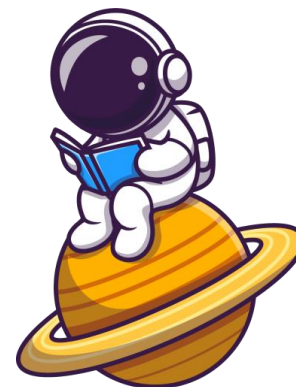
DCBA

CBA

BA

A

17. Mostrar un contador de 5 dígitos. En lugar del dígito 3 se debe mostrar E.





El futuro digital  
es de todos

MinTIC

# Git y Github







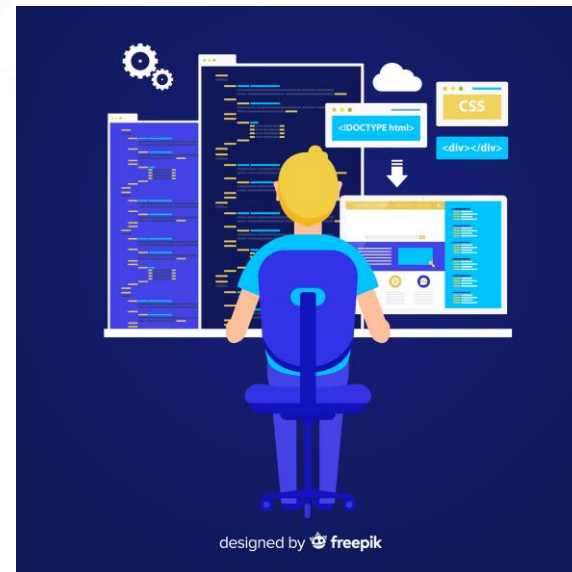
# Comandos que debemos conocer de Git

- `git init`: Inicializa un repositorio en un directorio local.
- `git clone`: Descargar el código fuente existente desde un repositorio remoto.
- `git status`: Da toda la información necesaria sobre la rama actual.
- `git add`: Incluir los cambios de los archivos en tu siguiente commit.
- `git commit`: Establecer un punto de control en el proceso de desarrollo al cual puedes volver más tarde si es necesario.
- `git push`: Envía tus commits al repositorio remoto.



# Ejercicio

- Iniciar un repositorio en el proyecto de los ejercicios de la clase 2
- Agregar el archivo `.gitignore` para excluir del commit las carpetas `target/` y `.settings/`
- Hacer commit al repositorio local
- Crear repositorio en GitHub
- Subir los cambios locales a GitHub
- Realizar un cambio a los archivos locales
- Volver a hacer el commit local y subir cambios a Github



[Vector de Negocios](https://www.freepik.es/fotos-vectores-gratis/negocios) creado por freepik - [www.freepik.es](https://www.freepik.es)



# Para la próxima sesión...

- Terminar los ejercicios que no se terminaron... (si aplica)
- Subir a Github los ejercicios terminados
- Revisar el material
  - Introducción a UML
  - Ver videos:
    - Curso Git, GitHub, VSCode

